

Visual Servoing in Presence of Non-Rigid Motion

D Santosh Kumar, C V Jawahar
Center for Visual Information Technology
International Institute of Information Technology
Hyderabad 500032, India
{santosh@students.,jawahar@}iiit.ac.in

Abstract

Most robotic vision algorithms have been proposed by envisaging robots operating in industrial environments, where the world is assumed to be static and rigid. These algorithms cannot be used in environments where the assumption of a rigid world does not hold. In this paper, we study the problem of visual servoing in presence of non-rigid objects and analyze the design of servoing strategies needed to perform optimally even in unconventional environments. We also propose a servoing algorithm that is robust to non-rigidity. The algorithm extracts invariant features of the non-rigid object and uses these features in the servoing process. We validate the technique with experiments and demonstrate the applicability.

1. Introduction

Visual servoing [8] describes a broad class of navigation problems in which a robot is positioned with respect to a target using computer vision as the primary feedback sensor. It is achieved by processing the visual feedback (2D features or 3D pose or both) and minimizing an appropriate error function.

Several algorithms [1, 3, 7, 10] have been proposed to perform the task of visual servoing; however, much of the research until now focuses only on rigid objects. As far as we are aware, the problem of visual servoing in presence of non-rigid objects has not been addressed in literature. The challenge with non-rigidity is that the motion planned based on the image-clues extracted at current time instant might not be valid in the next time instant as the object undergoes a change in its pose. This problem is different from that of servoing with respect to a moving object, where all the features that are being tracked undergo the same or predictable transformations in the image space. Whereas, for a deforming object, each feature point transforms differently and thus generates different control signals at the same time instant, thereby destabilizing the motion planning process.

Many of the living forms that we come across everyday

are non-rigid [2] in nature. For example, the articulated human motion; the deforming nature of the beating heart, flying birds and moving aquatic animals; the near-rigid nature of the human face etc. If robots are to be used in presence of such objects, we need servoing algorithms that are adaptable to non-rigidity. The work done by [7] and [10] deals with only moving rigid object visual servoing. Recently in [4], Cavusoglu *et al.* proposed a method to handle non-rigidity by canceling the relative motion between the non-rigid object (heart) and the camera by using a model predictive controller so as to facilitate easy manipulation of the object. In this paper, we address the issue of positioning of the end-effector of a robot with respect to a non-rigid object. Often, the positioning task is a pre-requisite for the manipulation task.

As a non-rigid object undergoes a persistent change in its pose, it is preferable to extract pose-invariant features of the object and use them in the servoing process. As a first attempt towards analyzing this problem, we investigate the various issues that need to be considered while solving it and also propose one simple yet an effective visual servoing algorithm to handle deformations. The algorithm is based on a not so strict assumption that any non-rigid motion can be approximated to a generalized quasi-periodic motion. And for a quasi-periodic motion, gross measures such as the moving average are sufficient to reveal its general stationary characteristics, provided the change in pose can be compensated. By using the recent results of multiple-view geometry, we efficiently extract such static features of the non-rigid object, which enable us to use the conventional visual servoing techniques even in this case.

2. Background

In image-based visual servo control, the error signal e is defined directly in terms of image feature parameters as $e(S) = S - S^*$, where S and S^* are the current and the desired feature parameters respectively. By differentiating

this error function with respect to time, we get:

$$\frac{de}{dt} = \frac{dS}{dt} = \left(\frac{\partial S}{\partial r}\right) \frac{dr}{dt} = L_S V, \quad (1)$$

where $V = (v^T, \omega^T)^T$ is the camera screw, r is the camera pose and L_S is the interaction matrix [8]. It relates the motion of the features in the image space to the camera motion in the Cartesian space. The main objective of the visual servoing process is to minimize the error function $e(S)$. For exponential convergence, we use a proportional control law i.e., $\frac{de(S)}{dt} = -\lambda e(S)$. By substituting this in (1), the required velocity of the camera can be computed as $V = -\lambda L_S^+ e(S)$, where L_S^+ is the pseudo-inverse of the interaction matrix and λ is a scale factor. In case of a moving object, the expression for V will get modified as (see [8])

$$V = -\lambda L_S^+ e(S) - L_S^+ \widehat{\frac{\partial e}{\partial t}}, \quad (2)$$

where $\widehat{\frac{\partial e}{\partial t}}$ represents the object motion model. The image-based visual servoing algorithm has gained prominence in literature due to its robustness to depth estimation errors.

Note that the above analysis assumes the objects to be rigid. Though non-rigid objects have not been studied in the visual servoing framework, they have been well analyzed in computer vision in the context of non-rigid tracking and 3D reconstruction problems. Most of these algorithms employ models like the *linear subspace model*, *appearance manifold* [11], *kinematic chains* [5, 9] etc in order to maintain a representation of the non-rigid object and also to infer its internal motion model. Such models allow prediction of the future states of the object, given its current state. Prediction is necessary in case of manipulation tasks where the end-effector needs to interact with the deforming object. For the positioning task, only an approximate pose of the object is sufficient to align the camera with respect to the object. Any error resulting due to this approximation can be corrected by using the visual feedback.

3. Visual Servoing in Presence of Deformations

Our problem of visual servoing is to position the end-effector of the robot with respect to a deforming object. Several strategies could be conceived to solve this problem. One possible solution is to use clues from object appearance to aid the positioning task. Alternatively, one could gain a stable representation of the object from its motion characteristics, which can be used to calibrate and move the end-effector. Another solution is to extract some pose-invariant features of the non-rigid object and use these features in the servoing process. Here we adopt a relatively simple strategy to solve this problem.

Any servoing algorithm that computes an optimal path for the robot to traverse from the initial position to the desired position, can be divided into two basic steps. After deciding on an appropriate feature to represent the non-rigid object, the following steps need to be repeated until the desired position is reached.

- Compute the error estimate between the current and the desired feature vector i.e., $S - S^*$
- Using a proportional control law (see Section 2), move in the direction that minimizes the above error

It is crucial to have a good feature vector S to represent the object state and thereby to perform the servoing task. However for non-rigid objects as the object pose continuously changes, we cannot obtain a feature vector from a single image that characterizes its state as it might lead to an unstable end-effector trajectory. For instance, a surgical robot positioning itself only using the current image of the heart, will oscillate corresponding to the systole and diastole states of the heart during its motion. Moreover, if only a single image is used to describe the desired position S^* , it will lead to oscillations of the manipulator even after the goal position is reached. One possible solution to circumvent this problem would be to use the sequence of images representing all states of the object to perform the servoing task. This will generate better control signals as it takes into account the object deformations. However, a more efficient solution could be to extract gross features from those sequence of images by exploiting the stationary signal property of the quasi-periodic object deformations.

At each pose, a sequence of images (I_1, I_2, \dots, I_n) are acquired by the camera, capturing the deformations of the non-rigid object (where n depends on the periodicity of the deformations). Then, m point features are extracted from each image I_i to obtain a feature vector $F_i^t = [f_1 f_2 \dots f_m]^T$, where each feature f_k corresponds to the projection of a 3D point on the surface of the non-rigid object at time instant t . The invariant feature G^t corresponding to the image sequence is computed as a function of all the feature vectors.

$$G^t = f(F_1^t, F_2^t, \dots, F_n^t) \quad (3)$$

However, the camera can use its previous knowledge and predict the next gross appearance of the deforming object. Thus at each position, only a single image is captured by the camera. The rest of the deformations of the object are obtained by projecting the past images into the new pose. Hence, the gross feature at time t can be predictively calculated as a function of the features extracted from the current image and the previous images as

$$G^t = f(g(F^{t-n}, \dots, F^{t-2}, F^{t-1}), F^t), \quad (4)$$

where F^{t-i} denotes the feature vector obtained from the image captured at camera pose r_{t-i} , F^t denotes the

feature vector extracted from the current image and $g(\cdot)$ is a function that projects the past image features into the current view as explained below.

Projection of Images Given two camera matrices M_1 and M_2 and an image I_1 taken from the pose corresponding to M_1 , the task is to obtain the image I_2 as seen from the pose corresponding to M_2 using a partial 3D model of the world that is assumed to be available in the image-based visual servoing process [8]. Let $M_1 = K_1[R_1 t_1]$ and $M_2 = K_2[R_2 t_2]$ where K_i denotes the intrinsic parameter matrix and $[R_i t_i]$ denotes the pose (rotation and translation). Let p_1 represent a point in I_1 and Z_1 be an estimate of the depth of the corresponding 3D point P_1 in the camera frame given by the 3D model. The coordinates of point P_1 are given by $P_1 = K_1^{-1}Z_1p_1$ (see [6]).

Now to obtain the 3D coordinate of this point in the other camera frame, we have

$$P_2 = [R_2 t_2][R_1 t_1]^{-1}P_1. \quad (5)$$

Finally, the image coordinates of the point with respect to the second camera are given by $p_2 = \frac{1}{Z_2}K_2P_2$. Thus by projecting the past image features into the new view, all the deformations of the object as viewed from the current camera pose can be obtained. These features are used to compute the representative feature of the non-rigid object.

4. Results and Discussions

In our experiments, we consider a set of four 3D points on the surface of a non-rigid object being observed by a perspective camera. The projection of the points onto the image are considered as features. The deformations of the non-rigid object are simulated by using a sinusoidal motion perturbed by a white gaussian noise. This makes the object undergo quasi-periodic deformations. The task of positioning a camera with respect to the non-rigid object is studied.

The basic implementation of the proposed algorithm can be summarized into the following steps.

1. In an offline step, acquire images from the desired final position. Extract feature S^* from these images using (3)
2. Acquire a sequence of images from the current camera pose and calculate S , again using (3)
3. Compute the velocity screw V using (2)
4. Using V , move the end-effector to the new pose and capture a new image
5. Project the features in the old images into the new pose as discussed in the latter half of Section 3
6. Using the features extracted from the old images and the current image, extract S using (4)
7. Repeat steps 3 to 6 until $e(S) < \theta$

We demonstrate our results using two kinds of features namely the mean and the eigen feature. Such features have been used in computer vision to capture statistical regularities and redundancies shared by a set of images. In the first method, all the feature vectors $F_i, i = 1, 2, \dots, n$ obtained from the n images are averaged to get the mean feature vector M . In this context, the mean operator corresponds to the function $f(\cdot)$ and M represents the invariant feature G^t . This feature is used as the input S to the servoing algorithm. As explained in Section 3, the new mean is predictively calculated using the past image features.

In the second method, the eigen feature of the sequence of image features is computed and used as the input feature to the servoing process. This feature indicates the direction in which the point features undergo maximum variation in the image space. In estimating the camera screw from (2), the line Jacobian is used to compute V since the eigen feature can be conceived of as a line segment in the image.

In comparison to classical image-based visual servoing, the use of above features generated a stable camera trajectory. Fig. 1a shows the velocity screw obtained in the case of the classical servoing algorithm. As only a single image is used, the generated control instructions resulted in an oscillatory camera trajectory (see the jagged behavior of the velocity screw in Fig. 1a as compared to that of Fig. 1b), affecting the stability of the robot. Also the system settled in a critically stable condition as the oscillations of the end-effector continued even after the desired position was reached.

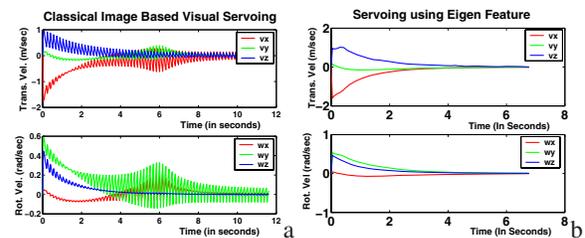


Figure 1. Velocity Screw (a) Oscillatory behavior (features only from single image are used) (b) Smooth convergence

We also tested the use of image sequence as an input feature to the servoing algorithm. In this case, the feature S was the entire sequence of features extracted from each image in the sequence i.e., $S = [F_1, F_2, \dots, F_n]^T$. As a large set of features are considered, minimizing the error difference between the current and the desired feature vector consumed more time and led to slower convergence of the error function (compare the error norm and the time taken in Fig. 2a to that of Fig. 2b).

Stability and Robustness We empirically verify the

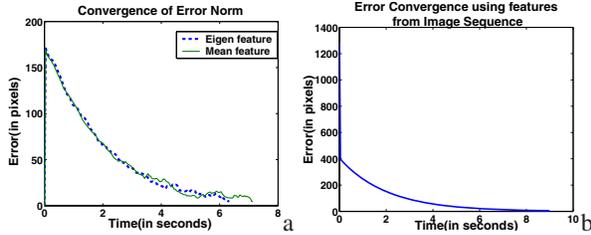


Figure 2. Error norm in case of (a) Eigen and mean feature (b) Image sequence

robustness of our approach to calibration errors. Often due to system calibration errors, the obtained image measurements are corrupted. The current image will be corrupted by the error in the current camera pose whereas the projected images will be corrupted by the error in the previous camera poses. This leads to the incorrect computation of the error function as $e'(S) = e(S) + \Delta e$ where $e(S)$ is as defined in Section 2 and Δe is the measurement error obtained due to the corresponding error in pose. As a result, the velocity screw changes to $V' = V + \Delta V$ where V is as given in (2) and ΔV is the velocity command generated to compensate the corresponding measurement error Δe . This ΔV accommodates the uncertainty in pose and thus ensures the convergence of error to zero. The stability of the algorithm was tested by introducing noise into the camera extrinsic parameters. The noise level was varied and the performance of the algorithm was studied. In Fig. 3a, it is observed that in spite of the calibration errors (a 10% gaussian noise was added), the convergence of velocity screw is achieved. This supports our analysis that uncertainty in pose is corrected by the visual feedback rather than getting propagated due to the prediction step.

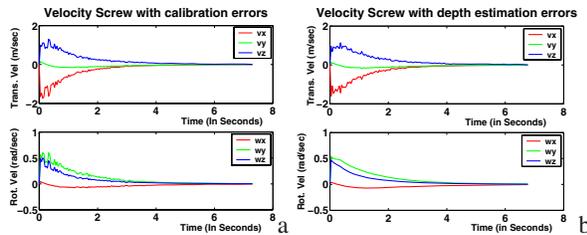


Figure 3. Convergence of velocity screw observed even in presence of errors (noise)

One of the crucial steps in the algorithm is the prediction step that assumes the depth of the feature points. The robustness of the algorithm to depth estimation errors was tested by introducing noise into the depth values. Fig. 3b shows the velocity screw obtained when the estimated depth was corrupted by 15% gaussian noise. This experiment demonstrated that minor error in the depth values can be accommodated by our algorithm.

5. Conclusion

Existing servoing schemes cannot adapt to the non-rigidity present in domestic environments. In this paper, we studied the issues involved in designing servoing strategies with optimal performance in presence of non-rigid objects. We also proposed a novel image-based visual servoing algorithm, which efficiently extracted gross features from sequence of images and used them in the servoing process. This algorithm generated a stable camera trajectory unlike the classical servoing algorithms. In future, we plan to improve the algorithm by considering other invariant features which better characterize the non-rigid object motion.

References

- [1] J. Adachi and J. Sato. Uncalibrated visual servoing from projective reconstruction of control values. *International Conference on Pattern Recognition*, 4:297–300, August 2004.
- [2] J. K. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Non-rigid motion analysis: articulated and elastic motion. *Computer Vision and Image Understanding*, 70(2):142–156, May 1998.
- [3] J. P. Barreto, J. Batista, and H. Araujo. Model predictive control to improve visual control of motion: Applications in active tracking of moving targets. *International Conference on Pattern Recognition*, 4:4732–4735, September 2000.
- [4] M. C. Cavusoglu, J. Rotella, W. S. Newman, S. Choi, J. Ustin, and S. S. Sastry. Control algorithms for active relative motion cancelling for robotic assisted off-pump coronary artery bypass graft surgery. *International Conference on Advanced Robotics*, pages 431–436, July 2005.
- [5] F. Chaumette, E. Marchand, and A. Comport. Object-based visual 3d tracking of articulated objects via kinematic sets. *IEEE Workshop on Articulated and Non-Rigid Motion, CVPRW*, 1:2–9, June 2004.
- [6] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.
- [7] K. Hashimoto, K. Nagahama, T. Noritsugu, and M. Taka-iawa. Visual servoing based on object motion estimation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1:245–250, October 2000.
- [8] S. A. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, October 1996.
- [9] T. Kanade, D. D. Morris, and J. Rehg. Ambiguities in visual tracking of articulated objects using two- and three-dimensional models. *International Journal of Robotics Research*, 22(6):393–418, June 2003.
- [10] E. Malis and S. Benhimane. A unified approach to visual tracking and servoing. *Robotics and Autonomous Systems*, 52(1):39–52, July 2005.
- [11] A. Shashua, A. Levin, and S. Avidan. Manifold pursuit: A new approach to appearance based recognition. *International Conference on Pattern Recognition*, 3:590–594, August 2002.